UD 4. MODELO FÍSICO DDL

- ¿Qué veremos la 1ª semana?
  - UD 4 MODELO FÍSICO DDL
    - Creación de bases de datos
    - Creación de tablas
    - Restricciones
    - Modificación de elementos
    - Borrado de tablas
  - Actividades propuestas (no evaluables)
    - Boletín A: ejercicios sobre modelado físico DDL

- ¿Qué veremos la 2ª semana?
  - Soluciones Boletín A
  - Actividades propuestas (no evaluables)
    - Boletín B: ejercicios sobre modelado físico DDL

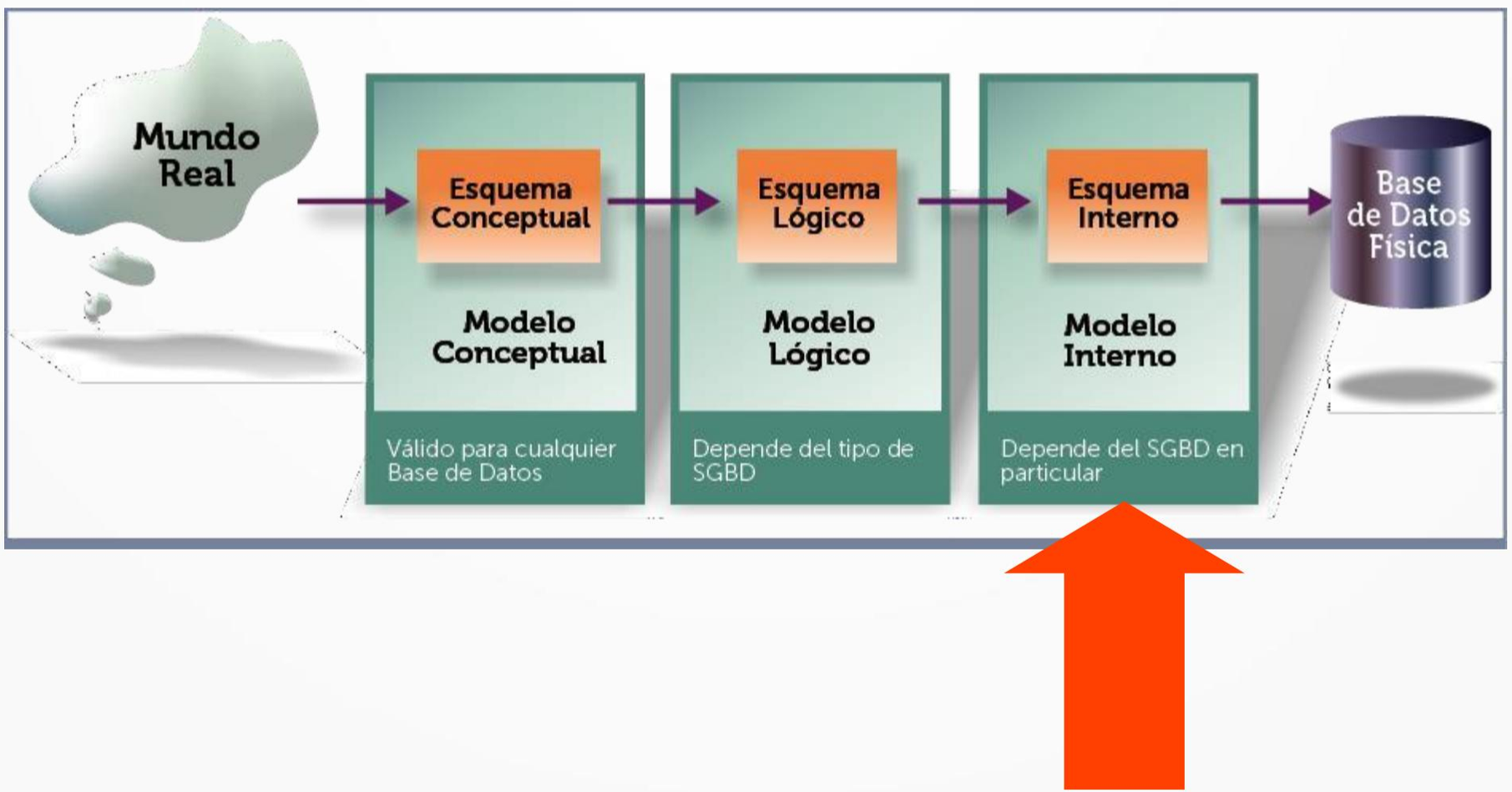
MODELO FÍSICO DDL

Para poder realizar el diseño físico de una BD (implementar la BD) lo primero es tener un software que te permita hacerlo y, también, familiarizarse con el uso del mismo.

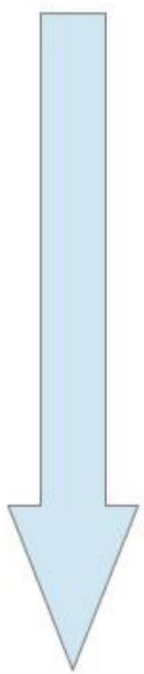
Como sabéis trabajaremos con SGBD MySQL o MariaDB; como cliente usaremos WorkBench. Éste es el software con el que se harán los ejercicios y los exámenes.

En la Unidad, al principio, tiene indicaciones de cómo instalar el software que necesita.
En caso de tener LINUX, puede usar el software incluido en la carpeta SOFTWARE Y,
si no hay otro remedio, si no desea realizar la instalación, puede usar la VM del enlace.

4.1 Modelado físico



4.1 Modelado físico

NIVEL DE ABSTRACCIÓN	FASE DEL MODELADO	DEPENDENCIA CON EL SGBD	TÉCNICA MÁS USADA
ALTA	REALIDAD		
	Conceptual	Ninguna dependencia *Modelo 100% teórico*	MODELO E-R DE PETER CHEN
	Lógico	Dependencia con tipo de SGBD *Relacional, jerárquico, objetos, no-relacional ...*	MODELO RELACIONAL DE EDGAR CODD
	Físico	Con el SGBD concreto *Relacional: MariaDB, Oracle, MySQL, PostgreSQL...* *No-relacional: MongoDB, Cassandra...*	Especificaciones de cada SGBD
BAJA	SISTEMA DE INFORMACIÓN		

4.1 Modelado físico

381 systems in ranking, November 2021

Rank			DBMS	Database Model	Score		
Nov 2021	Oct 2021	Nov 2020			Nov 2021	Oct 2021	Nov 2020
1.	1.	1.	Oracle	Relational, Multi-model	1272.73	+2.38	-72.27
2.	2.	2.	MySQL	Relational, Multi-model	1211.52	-8.25	-30.12
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	954.29	-16.32	-83.35
4.	4.	4.	PostgreSQL	Relational, Multi-model	597.27	+10.30	+42.22
5.	5.	5.	MongoDB	Document, Multi-model	487.35	-6.21	+33.52
6.	6.	↑ 7.	Redis	Key-value, Multi-model	171.50	+0.15	+16.08
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model	167.52	+1.56	+5.90
8.	8.	8.	Elasticsearch	Search engine, Multi-model	159.09	+0.84	+7.54
9.	9.	9.	SQLite	Relational	129.80	+0.43	+6.48
10.	10.	10.	Cassandra	Wide column	120.88	+1.61	+2.13

Más información sobre SGBD en https://db-engines.com/en/ranking En este módulo trabajaremos con MySQL con el cliente WorkBench

4.1 Modelado físico

Las diferentes operaciones que se pueden realizar en una BD se clasifican en si esas operaciones se refieren a los datos ya las metadas

entendidos estos últimos como las “estructuras” en las que almacenamos o clasificamos los datos.

Con esto, podríamos tener esta clasificación:

•Operaciones sobre los METADATOS:

• El DDL (Data Definition Language) se encarga de la creación, modificación y eliminación de los objetos de la BD (de los metadatos)

•Operaciones sobre los DATOS:

• El DML (Data Manipulation Language) se encarga de la creación, modificación y eliminación de los DATOS.

• El DQL (Data Query Language) se encarga de la consulta o lectura de los DATOS.

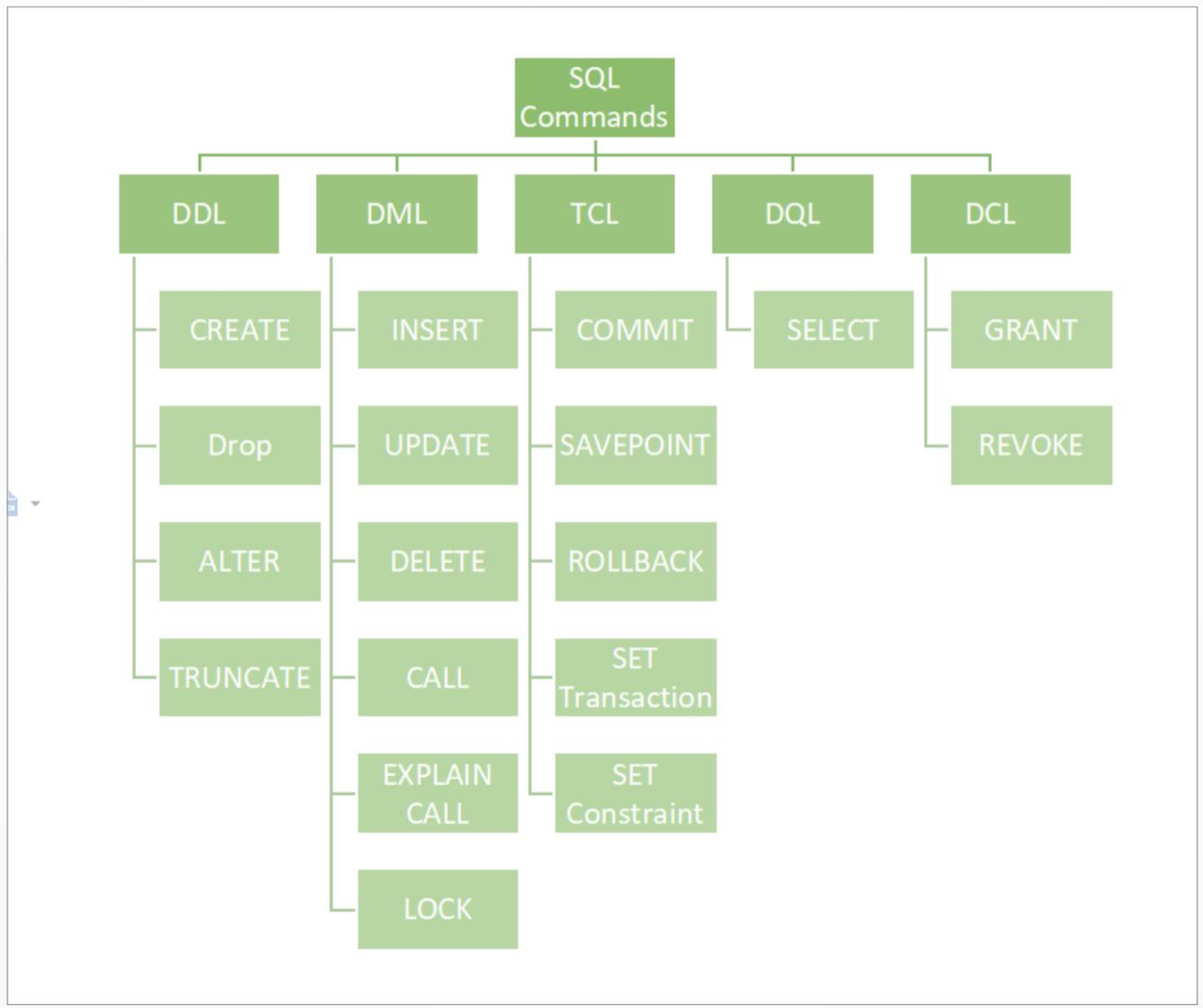
Veremos en este tema el DDL y en los siguientes el DML y DQL, respectivamente.

4.1 Modelado físico

SQL (por sus siglas en inglés Structured Query Language)lenguaje de consulta estructurada es un lenguaje de dominio específico utilizado en programación, diseñado para administrar, y recuperar información de SGBDR.

En la actualidad SQL es el estándar de la mayoría de los SGBDR y, aunque la diversidad de añadidos particulares que incluyen las diferentes implementaciones comerciales del lenguaje es amplia, el soporte al estándar SQL-92 es general y muy amplio.

4.1 Modelado físico



4.2 Creación de bases de datos

MySQL permite crear diferentes bases de datos en un mismo servidor (también se pueden denominar Schemas

Hay decenas de tutoriales y videotutoriales para instalar de forma sencilla MySQL en tu equipo.

Tienes indicaciones en el archivo 'Instalación SGBD'

4.2 Creación de bases de datos

Veamos cómo se crea una BD en MySQL.

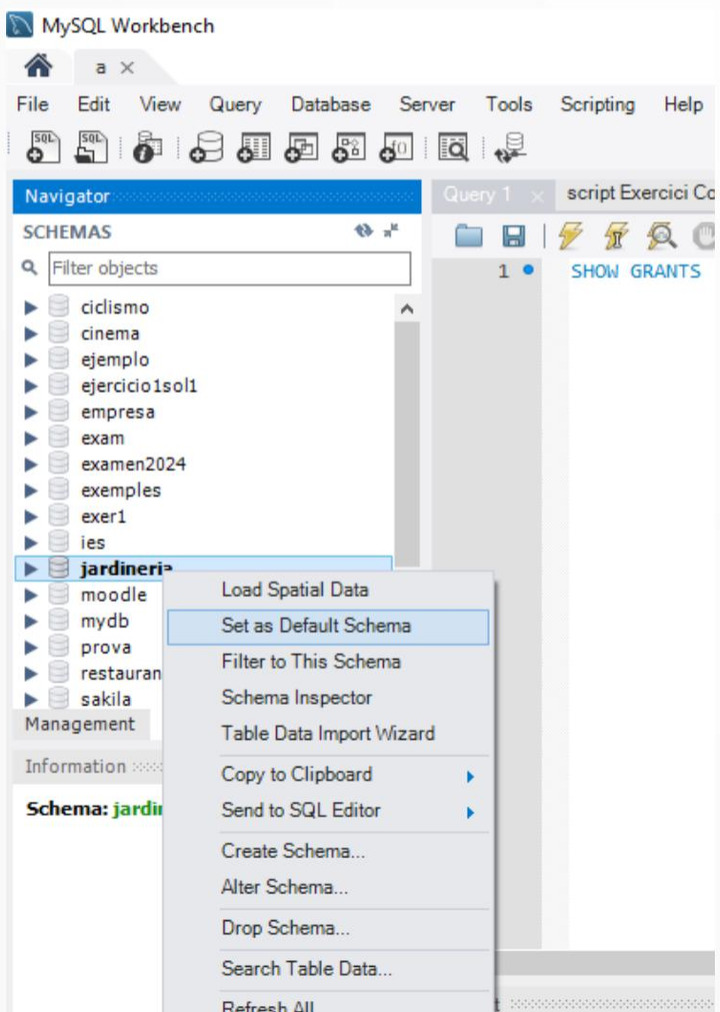
```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] nombre_base_de_datos

[especificación_create [, especificación_create] ...]

especificación_create:

[DEFAULT] CHARACTER SET juego_de_caracteres |

[DEFAULT] COLLATE nombre_de_colación
```

En el comando es CREATE, después se puede especificar DATABASE o SCHEMA, es indiferente utilizar cualquiera de las dos para crear la base de datos. Cuando se pueden elegir un elemento entre varios éstos van entre llaves { } y separados por un pipe |.

Por otro lado, los elementos que van entre corchetes son opcionales, pueden ir o no, por ejemplo es opcional poner IF NOT EXIST, es decir que cree la base de datos si no existe ya. Por último, puedes observar la especificación del conjunto de caracteres a utilizar o bien la forma de tratar ese conjunto de caracteres.

4.2 Creación de bases de datos

Si creemmésd'unaBDivolemcanviarentrel'unail'altrapodemfer-hod'aquestamanera:



- El nombre de una tabla sigue la siguiente nomenclatura:

  NombreEsquema.NombreTabla

- Si trabajaremos con una determinada BD

  (esquema), lo más cómodo es designar esa BD como esquema por defecto. Con esto nos ahorramos escribir el nombre del esquema cada vez.

- Se puede realizar también con la instrucción:

  USO NomEsquema;

- Hasta que no se indique lo contrario, el SGBD trabajará con el esquema elegido (si lo hubiere)

4.3 Creación de tablas

Sentencia básica para crear tablas

```
CREAR TABLA departamentos
( codDpto VARCHAR(10) CLAVE PRIMARIA,
Nombre VARCHAR(30) NO NULL,
Ubicacion VARCHAR(30) UNIQUE );
```

```
CREATE TABLE númerotabla
( campo1 tipo restricción,
...
campoN tipo restricción );
```

Tabla Departamentos.

CodDpto	Nombre	Ubicación
INF	Informática	Planta sótano U3
ADM	Administración	Planta quinta U2
COM	Comercial	Planta tercera U3
CONT	Contabilidad	Planta quinta U1
ALM	Almacén	Planta baja U1

4.3 Creación de tablas

Sentencia básica para crear tablas

```
CREAR TABLA departamentos
( codDpto VARCHAR(10) CLAVE PRIMARIA,
Nombre VARCHAR(30) NO NULL,
Ubicacion VARCHAR(30) UNIQUE );
```

PodemllançarordresalaBDdequatremaneres:

1)Introducción a oraciones sin sentido en la consola

2) Usando un IDE BBDD

3) Cargando comandos xiuambels

4)MedianteunprogramaoAPP(porejemploWorkBench)

4.3 Creación de tablas

Tipos de campos

Tipos numéricos de los SGBDR más comunes:

MySQL Data Type	Oracle Data Type
BIGINT	NUMBER(19, 0)
BIT	RAW
BLOB	BLOB, RAW
CHAR	CHAR
DATE	DATE
DATETIME	DATE
DECIMAL	FLOAT (24)
DOUBLE	FLOAT (24)
DOUBLE PRECISION	FLOAT (24)
ENUM	VARCHAR2
FLOAT	FLOAT
INT	NUMBER(10, 0)
INTEGER	NUMBER(10, 0)
LONGBLOB	BLOB, RAW
LONGTEXT	CLOB, RAW
MEDIUMBLOB	BLOB, RAW
MEDIUMINT	NUMBER(7, 0)
MEDIUMTEXT	CLOB, RAW
NUMERIC	NUMBER
REAL	FLOAT (24)
SET	VARCHAR2
SMALLINT	NUMBER(5, 0)
TEXT	VARCHAR2, CLOB
TIME	DATE
TIMESTAMP	DATE
TINYBLOB	RAW
TINYINT	NUMBER(3, 0)
TINYTEXT	VARCHAR2
VARCHAR	VARCHAR2, CLOB
YEAR	NUMBER

Tipos relativos a fechas y horas:

MySQL	Oracle
DATE	DATE
DATETIME	DATE
TIMESTAMP	DATE
TIME	DATE
YEAR	NUMBER

Tipos String:

MySQL	Oracle
BLOB	RAW, BLOB
CHAR(m)	CHAR
ENUM (VALUE1, VALUE2, ...)	
LONGBLOB	RAW, BLOB
LONGTEXT	RAW, CLOB
MEDIUMBLOB	RAW, BLOB
MEDIUMTEXT	RAW, CLOB
SET (VALUE1, VALUE2, ...)	
TEXT	VARCHAR2, CLOB
TINYBLOB	RAW, BLOB
TINYTEXT	VARCHAR2
VARCHAR(m)	VARCHAR2, CLOB

```
CREATE TABLE nombre
( campo1 tipo restricción,
...
campoN tipo restricción );
```

4.4 Restricciones

Restricciones

CREATE TABLE númerotabla
(campo1 tipo

restricción, ...); • PRIMARY KEY: establece ese atributo como la clave primaria de la tabla (CP).

• PRIMARY KEY lista_columnas: establecer como CP un conjunto de atributos.

• FOREIGN KEY: define una clave externa de la tabla respecto a otra tabla (CA).

• UNIQUE: sin valores repetidos e incluye Valor No Nul (VNN).

• NOT NULL: evita que se introduzcan tuplas con valor NULL para ese atributo (VNN).

• CHECK (condición): permite establecer condiciones que deben cumplir atributos.

• DONDE DELETE CASCADE|SET NULL: especifica que se mantenga automáticamente la integridad referencial borrando o poniendo a nulos los valores de la clave externa correspondientes a un valor borrado de la mesa referenciada (tabla padre).

4.4 Restricciones

Restricciones

Existen dos formas de incluir las restricciones: •por columna •por nombre de restricción

```
CREATE TABLE númerotabla
( campo1 tipo restricción,
...
campoN tipo restricción );
```

```
CREAR TABLA departamentos
( codDpto VARCHAR(10) CLAVE PRIMARIA,
Nombre VARCHAR(30) NO NULL,
Ubicacion VARCHAR(30) UNIQUE );
```

4.4 Restricciones

Restricciones

Existen dos formas de incluir las restricciones: •por columna •por nombre de restricción

```
CREATE TABLE númerotabla
( campo1 tipo restricción,
...
campoN tipo restricción );
```

Eso obliga a que la columna tenga que tener obligatoriamente un valor para que sea almacenado el registro. Se puede especificar durante la creación (o modificación) del campo añadiendo la palabra NOT NULL tras el tipo:

```
CREATE TABLE cliente (
dni VARCHAR2(9) NOT NULL);
```

En ese caso el nombre lo coloca la propia base de datos (en el caso de Oracle el nombre sería algo como SY002341 por ejemplo). Para especificar nosotros el nombre se usa:

```
CREATE TABLE cliente(
dni VARCHAR2(9) CONSTRAINT cli_dni_nn NOT NULL);
```

Nuevamente hay dos formas de colocar esta restricción. Si no se especifica el nombre, este lo pondrá el sistema:

```
CREATE TABLE cliente(
dni VARCHAR2(9) UNIQUE);
```

O podemos indicar el nombre de la restricción de la siguiente manera:

```
CREATE TABLE cliente (
dni VARCHAR2(9) CONSTRAINT cli_dni_uk UNIQUE);
```

Si la restricción se refiere a varios campos, la forma sería:

```
CREATE TABLE alquiler (
dni VARCHAR2(9),
cod_pelicula NUMBER(5),
CONSTRAINT alq_dnicod_uk UNIQUE(dni, cod_pelicula));
```

4.4 Restricciones

Restricciones

Existen dos formas de incluir las restricciones: •por columna •por nombre de restricción

```
CREATE TABLE númerotabla
( campo1 tipo restricción,
...
campoN tipo restricción );
```

Si la clave está formada por un solo campo basta con definirla de la siguiente manera:

```
CREATE TABLE cliente (
dni VARCHAR(9) PRIMARY KEY,
nombre VARCHAR(50));
```

O, poniendo un nombre a la restricción:

```
CREATE TABLE cliente (
dni VARCHAR(9) CONSTRAINT cli_dni_pk PRIMARY KEY,
nombre VARCHAR(50));
```

Si la clave está formada por más de un campo:

```
CREATE TABLE alquiler (
dni VARCHAR(9),
cod_pelicula NUMBER(5),
CONSTRAINT alq_pk PRIMARY KEY(dni,cod_pelicula));
```

La forma de indicar una clave ajena (aplicando una restricción de integridad referencial) es haciendo referencia a la tabla y el campo/s con la que se relaciona a través de la palabra clave REFERENCES:

```
CREATE TABLE alquiler (
dni VARCHAR2(9) CONSTRAINT alq_dni_fk REFERENCES clientes(dni),
cod_pelicula NUMBER(5) CONSTRAINT alq_cod_fk REFERENCES peliculas(cod),
CONSTRAINT alq_pk PRIMARY KEY(dni,cod_pelicula));
```

Que significa que el campo dni de la tabla alquiler se relaciona con el campo dni de la tabla clientes.

Si el campo al que se hace referencia es la clave principal, se puede obviar el nombre del campo:

```
CREATE TABLE alquiler (
dni VARCHAR2(9) CONSTRAINT alq_dni_fk REFERENCES clientes,
cod_pelicula NUMBER(5) CONSTRAINT alq_cod_fk REFERENCES peliculas,
CONSTRAINT alq_pel_pk PRIMARY KEY(dni,cod_pelicula));
```

4.4 Restricciones

Restricciones

Existen dos formas de incluir las restricciones:

- por columna
- por nombre de restricción

•

Nota: En MariaDB la restricción de columna de Clave Aliena no está implementada (REFERENCIAS...) Por lo que si no definimos las Claves Ajenas como restricciones de tablas, no tendremos ninguna efectos.

```
CREATE TABLE EMPLEADOS (
        ID_EMP AUTO_INCREMENT PRIMARY KEY,
        NOM_EMP VARCHAR(12),
        ID_JEFE INTEGER REFERENCES JEFES(ID_JEFE)
          ON UPDATE CASCADE ON DELETE RESTRICT,
        TARIFA_HR INTEGER,
        CHECK (TARIFA_HR>0) )
-- Alternativas PK y FK (1° campos 2° constraints):
CREATE TABLE EMPLEADOS (
        ID_EMP AUTO_INCREMENT,
        ID_JEFE INTEGER,
        CONSTRAINT PK_EMP PRIMARY KEY (ID_EMP),
        CONSTRAINT FK_JEFE FOREIGN KEY (ID_JEFE)
          REFERENCES JEFES(ID_JEFE)
          ON UPDATE CASCADE ON DELETE RESTRICT,
        CONSTRAINT CHK_XX CHECK (TARIFA_HR>0))
```

4.4 Restricciones

AL ELIMINAR | AL ACTUALIZAR

Podemos especificar qué debe hacer el SGBD cuando intentamos borrar un registro cuya clave primaria (PK) está referenciada en otras tablas como parte de una clave ajena (FK).

Por ejemplo, qué hacer cuando borramos una asignatura en la que se han matriculado alumnos.

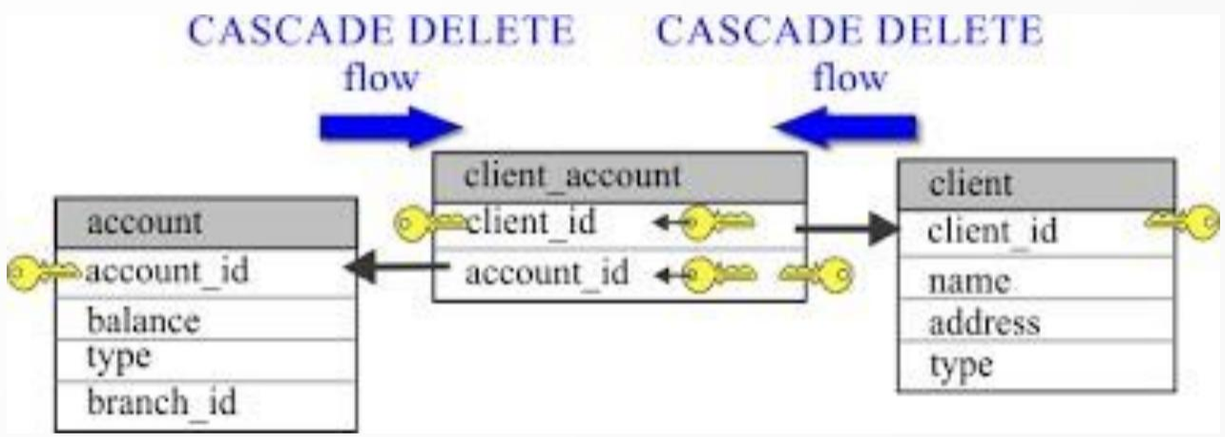
Tenemos varias opciones: •ON DELETE NO ACTION (la opción por defecto) • El SGBD nos daría un error y no nos dejaría borrar esa asignatura hasta que no “desmatriculáramos” a los alumnos. •DÓNDE DELETE CASCADE • Al borrar la asignatura borramos todos los registros donde aparezca la clave primaria de esa asignatura. •ONDELETE SET NULL • Al borrar la asignatura marcamos en NULO todos los campos donde aparezca la clave primaria de esa asignatura. •ON DELETE SET DEFAULT • Al borrar la asignatura rellenamos con el valor por defecto todos los campos donde aparezca la clave primaria de esa asignatura.

Y lo mismo con UPDATE cuando se cambie (renombre) la clave primaria de un registro que forma parte de una clave ajena.

4.4 Restricciones

AL ELIMINAR | AL ACTUALIZAR

```
CREATE TABLE alquiler(
dni VARCHAR(9),
cod_pelicula NUMBER(5),
CONSTRAINT alq_pk PRIMARY KEY(dni,cod_pelicula),
CONSTRAINT alq_dni_fk FOREIGN KEY (dni)
REFERENCES clientes(dni) ON DELETE SET NULL,
CONSTRAINT alq_pel_fk FOREIGN KEY (cod_pelicula)
REFERENCES peliculas(cod) ON DELETE CASCADE );
```



4.4 Restricciones

CONTROLAR

```
CREATE TABLE ingresos(
cod NUMBER(5) PRIMARY KEY,
concepto VARCHAR2(40) NOT NULL,
importe NUMBER(11,2),
CONSTRAINT ing_imp_min_ck CHECK (importe>0)
CONSTRAINT ing_imp_max_ck CHECK (importe<8000));
```

```
CREATE TABLE ingresos (
cod NUMBER(5) PRIMARY KEY,
concepto VARCHAR2(40) NOT NULL,
importe_max NUMBER(11,2),
importe NUMBER(11,2),
CONSTRAINT ing_imp_max_ck CHECK (importe<importe_max));
```

4.5 Modificación de elementos

CAMBIO DE CAMPOS

MySQL

```
✚ALTER TABLE EMPLEADOS ...
añade campo
  ADD [COLUMN] tipo INT [NOT NULL] [AFTER campo];
cambia solo tipo de un campo
  MODIFY [COLUMN] email VARCHAR(10) [NOT NULL];
cambia solo nombre de un campo
  MODIFY [COLUMN] email email2;
cambia tipo y nombre de un campo
  CHANGE [COLUMN] email correo VARCHAR(20) [NOT NULL];
elimina conlumna
  DROP [COLUMN] correo;
```

Ejemplo:

ALTER TABLE contactos

MODIFY ciudad ciudad_residencia;

4.5 Modificación de elementos

CAMBIO DE CAMPOS

Ejemplos prácticos de ALTER TABLE en MySQL

Comenzamos con un ejemplo para **renombrar y/o cambiar el nombre la tabla**:

```
1 ALTER TABLE nombre_tabla RENAME nombre_nuevo_tabla;
```

Cambiar el juego de caracteres de la tabla:

```
1 ALTER TABLE nombre_tabla CHARACTER SET latin1;
```

Cambiar el engine (motor) de almacenamiento:

```
1 ALTER TABLE nombre_tabla ENGINE = InnoDB;
```

Eliminar una columna de la tabla:

```
1 ALTER TABLE nombre_tabla DROP COLUMN nombre_columna;
```

Eliminar varias columnas de la tabla:

```
1 ALTER TABLE nombre_tabla DROP COLUMN nombre_columna, DROP COLUMN nombre_columna2;
```

Eliminar una clave primaria y clave externa (**FOREING KEY y PRIMARY KEY**):

```
1 #Eliminar clave primaria
2 ALTER TABLE nombre_tabla DROP PRIMARY KEY;
3 #Eliminar clave externa
4 ALTER TABLE nombre_tabla DROP FOREIGN KEY nombre_columna;
```

En el siguiente código vamos a **insertar una nueva columna al final de la tabla**:

```
1 ALTER TABLE nombre_tabla ADD fecha_nacimiento date;
```

Añadir una nueva columna después de otra:

```
1 ALTER TABLE nombre_tabla ADD nombre_columna VARCHAR(5) AFTER nombre_columna_anterior;
```

Añadir una nueva columna en la primera posición de la tabla:

```
1 ALTER TABLE nombre_tabla ADD nombre_columna VARCHAR(5) INT FIRST;
```

https://www.anerbarrena.com/alter-table-mysql-5050/

4.5 Modificación de elementos

AGREGAR/ELIMINAR RESTRICCIÓN

añade constraint

```
ADD CONSTRAINT fk_xx
      FOREIGN KEY (ID_JEFE) REFERENCES tabla(campo)..;
ADD CONSTRAINT chk_edad
      CHECK (EDAD BETWEEN 10 AND 99)
        OR (EDAD IN (20,25,30) OR (EDAD IS NULL);
ADD CONSTRAINT chk_tipo
      CHECK (TIPO IN ('INGENIERO', 'BECARIO');
```

drop constraint

```
DROP CONSTRAINT fk_xx;
```

Ejemplo:

ALTER TABLE contactos
ADD CONSTRAINT chkCiudad CHECK (ciutat="Madrid" or ciutat="València");

4.5 Modificación de elementos

AGREGAR/ELIMINAR RESTRICCIÓN

Más ejemplos:

https://www.cablenaranja.com/como-crear-y-manejar-constraints-en-mysql/

4.6 Borrado de tablas

BORRAR MESAS

TRUNCATE TABLE usuarios;

VACÍA LA MESA (BORRA SÓLO LOS DATOS

DROP TABLE usuarios;

=> BORRA LA MESA (BORRA LOS METADATOS Y, CON ESO, BORA TAMBIÉN LOS DATOS)

ACTIVIDADES

Consulta los ejercicios sugeridos que encontrarás en Aulas:

Ejercicios DDL Boletín A

Ejercicios DDL Boletín B

Comprender estas actividades no evaluables es esencial.

Intenta solucionar el problema utilizando los recursos que te facilitamos y la documentación ampliada que encontrarás en el Aula Virtual, acudiendo al foro de la unidad si tienes cualquier duda que no sepas resolver.